Discrete Optimization 10 (2013) 11–24

Contents lists available at SciVerse ScienceDirect

Discrete Optimization

journal homepage: www.elsevier.com/locate/disopt

A logarithmic method for eliminating binary variables and constraints for the product of free-sign discrete functions

Hao-Chun Lu

Department of Information Management, College of Management, Fu Jen Catholic University, No.510, Jhongjheng Rd., Sinjhuang City, Taipei County 242, Taiwan

A R T I C L E   I N F O

Article history:
Received 22 October 2010
Received in revised form 5 February 2012
Accepted 7 October 2012
Available online 31 October 2012

Keywords:
Discrete function
Free-sign variables
Mixed-integer non-linear programming

A B S T R A C T

In this paper, a logarithmic method was developed to solve optimization problems containing the product of free-sign discrete functions (PFDF). The current deterministic methods used to handle these problems are based on the concept of continuous variables; therefore, the methods always transform the original model into another programming model (e.g., DC programming, convex programming) and solve them with a commercial solver. As the nature of a discrete variable is quite different from that of a continuous one, developing a novel method to address the above mentioned problems is necessary. This study proposes a concise and efficient method that linearizes PFDF term into a set of linear inequalities directly without redundant transformation. Further, the proposed method only requires the logarithmic numbers of binary variables and constraints. Numerical examples demonstrate that the proposed formulation significantly outperforms current approaches.

1. Introduction

Many optimization problems contain the product of free-sign discrete functions (PFDF), such as Geometric Programming (GP), Generalized Geometric Programming (GGP), and Mixed-Integer Non-Linear Programming (MINLP). Applications based on the above programming methods are used in a wide variety of fields, such as computer-aided design [1], digital circuits [2], smoothing splines [3], and communication systems [4]. PFDF is a popular and fundamental component of optimization problems because discrete variables represent standardization benchmarks in industry (e.g., standard sizes of components, thicknesses of steel plates, diameters of pipes, lengths of springs, and elements in a competence set). The entire optimization problem containing the PFDF terms can be represented by Program 1:

Program 1. The PFDF problem

$$\text{Min} \sum_{p=1}^{T} D_p(\mathbf{y}) + \sum_{p=1}^{T'} F_p(\mathbf{x})$$

$$\text{s.t.} \sum_{p=1}^{T_w} D_{w,p}(\mathbf{y}) + \sum_{p=1}^{T'_w} F_{w,p}(\mathbf{x}) \le l_w, \quad w = 1, \ldots, s,$$

where

(i) $D_p(\mathbf{y}) = c_p \prod_{i=1}^{n} g_{p,i}(y_i)$ and $D_{w,p}(\mathbf{y}) = c_{w,p} \prod_{i=1}^{n} g_{w,p,i}(y_i)$ are the PFDF terms, $g_{p,i}(y_i)$ and $g_{w,p,i}(y_i)$ are the discrete functions with the free-sign discrete variables y_i;

E-mail addresses: bach0809@gmail.com, haoclu@gmail.com.

1572-5286/$ – see front matter © 2012 Elsevier B.V. All rights reserved.
doi:10.1016/j.disopt.2012.10.001

(ii) $\mathbf{y} = (y_1, y_2, \ldots, y_n)$ is a free-sign discrete variable vector, $y_i \in \{d_{i,1}, d_{i,2}, \ldots, d_{i,r_i}\}$ for $i = 1, \ldots, n$ and $d_{i,r_i} \in \mathbb{R}\ \forall i$;
(iii) $F_p(\mathbf{x})$ and $F_{w,p}(\mathbf{x})$ are either convex functions or linear functions, and $\mathbf{x} = (x_1, x_2, \ldots, x_s)$ is a positive continuous variable vector;
(iv) $\alpha_{p,i}, \alpha_{w,q,i}, c_p, c_{w,q},$ and l_w are constants.

In this study, the focus is on developing an exact linearization of the PFDF terms ($D_p(\mathbf{y})$ and $D_{w,p}(\mathbf{y})$) in Program 1. Other terms ($F_p(\mathbf{x})$ and $F_{w,p}(\mathbf{x})$) do not need any transformation because they are either convex functions or linear functions. If any non-convex signomial terms occur in $F_p(\mathbf{x})$ or $F_{w,p}(\mathbf{x})$, they can be convexified via many transformation techniques, such as exponential transformation, negative power transformation, or positive power transformation techniques; Lundell [5] provided an impressive overview of said techniques.

Many heuristic optimization algorithms have been developed to solve the PFDF problem. Xiong and Rao [6] proposed a hybrid approach that combines fuzzy nonlinear programming and a genetic algorithm to solve polynomial integer programs with mixed-discrete variables. Rosen and Harmonosky [7] improved the simulated annealing method for discrete variable optimization. Such heuristic algorithms can find feasible solutions within an acceptable time, but they cannot guarantee to achieve global optimization. Thus, this paper focuses on deterministic models for solving the optimization problems containing PFDF terms.

Currently, Floudas' methods [8–11] are the most popular deterministic approaches for solving GGP problems. These problems form a subclass of PFDF problems, but the variables are continuous. Floudas' method represents each term in a GGP problem as the difference between two signomial terms. Then, each signomial term is converted into a convex underestimator by an exponential transformation technique. Pörn et al. [12] also utilized the exponential transformation technique to propose a MINLP formulation for solving the discrete posynomial term. For simplicity, the discrete functions $g_i(y_i)$ in the PFDF term may be considered as power functions, i.e., $g_i(y_i) = y_i^{\alpha_i}$, and the following equation can be an example of a PFDF term:

$$D(\mathbf{y}) = c\prod_{i=1}^{n} y_i^{\alpha_i}, \quad y_i \in \{d_{i,1}, d_{i,2}, \ldots, d_{i,r_i}\}\forall i,\ \alpha_1 \le \cdots \le \alpha_m < 0 < \alpha_{m+1} \le \cdots \le \alpha_n. \tag{1}$$

The special form of the PFDF term in (1) is also a signomial term, which is a fundamental component in the GGP problem. The exponential transformation technique transforms $D(\mathbf{y})$ with $d_{i,r_i} > 0\ \forall i, r_i$ in (1) into the following equations:

$$D(\mathbf{y}) = c \cdot \exp\left(\sum_{i=1}^{n} \alpha_i z_i\right), \tag{2}$$

$$z_i = \ln y_i = \ln d_{i,1} + \sum_{k=2}^{r_j} u_{i,k}(\ln d_{i,k} - \ln d_{i,1}), \quad \sum_{k=2}^{r_j} u_{i,k} \le 1 \quad \text{for all } u_{i,k} \in \{0, 1\}. \tag{3}$$

Eq. (3) is a set of linear functions, and the exponentiation of linear functions in (2) is convex. Thus, Floudas' method can achieve finite ε-convergence to global optimization by successively refining a convex relaxation of a series of nonlinear convex optimization problems.

Remark 1. In linearizing the PFDF term in (1), the methods of Floudas [8–11] and Pörn et al. [12] require $\sum_{i=1}^{n}(r_i - 1)$ binary variables, $2n$ linear constraints, and n auxiliary non-negative continuous variables.

Recently, Pörn et al. [13] proposed an improved exponential transformation technique for a signomial term (the special form of the PFDF term) with continuous variables. They convexified the original non-convex signomial without introducing any additional signomials, and only applying exponential transformation to the related variables, instead of all the variables, in the non-convex signomial terms. Their method can be formulated as follows:

$$\text{(Case 1) } c > 0 : D(\mathbf{y}) \to c\prod_{j=1}^{m} y_j^{\alpha_j} \cdot \exp\left(\sum_{k=m+1}^{n} \alpha_k z_k\right), \quad \text{where } z_k = \ln y_k\ \forall k, \tag{4}$$

$$\text{(Case 2) } c < 0 : D(\mathbf{y}) \to c\prod_{j=1}^{m} z_j^{\left|\frac{-\alpha_j}{A}\right|} \cdot \prod_{k=m+1}^{n} z_k^{\frac{\alpha_k}{A}},$$

$$\text{where } A = \sum_{i=1}^{n} |\alpha_i|,\ z_j = y_j^{-A},\ j = 1, \ldots, m;\ z_k = y_k^{A},\ k = m+1, \ldots, n. \tag{5}$$

The methods of Floudas [8–11] and Pörn et al. [13] are applicable to optimization problems with positive continuous variables as they produce valid convex underestimators for the original optimization problems. However, these methods, which employ the exponential transformation technique, are not appropriate for the PFDF term as the variables in the PFDF term can be negative values. Additionally, the methods of Floudas [8–11] and Pörn et al. [12] produce a large number of binary variables that linearize the PFDF term in a piecewise manner. For solving mixed-integer signomial programming (MISP) problems, Lundell [5] proposed an efficient method, the SGO-algorithm, in his Ph.D. dissertation. The SGO-algorithm solves the MISP problems of the form as a sequence of convexified and overestimated MINLP problems and presents a preprocessing step to provide the transformations used for convexifying the non-convex signomial terms.

To resolve the difficulty of using the exponential transformation technique with free-sign variables, Pörn et al. [12] proposed a simple conversion, $y + \tau = e^z$, to convert the variables. However, this conversion requires numerous additional signomial terms and incurs excessive computational overhead. Under this method, the PFDF term $D(\mathbf{y})$ in (1) with $\alpha_i \in \mathbb{Z}^+ \forall i$ will be transformed into $\prod_{i=1}^{n}(\alpha_i + 1) - 1$ additional signomial terms and one constant. Moreover, dealing with a PFDF term with $\alpha_i \notin \mathbb{Z}^+$ is difficult. For instance, transforming the signomial term $y_1^{1/3} y_2^{-5}$, based on Pörn's translation, is not possible. Tsai and Lin [14] used the difference between two positive variables to solve the free-sign variable issue. Their method can be formulated as follows:

$$y_i = y_i^+ - y_i^-; \qquad y_i^\alpha = (y_i^+)^\alpha + (-1)^\alpha (y_i^-)^\alpha, \tag{6}$$

$$-y_i^- \le y_i \le Mu_i - y_i^-; \qquad M(u_i - 1) + y_i^+ \le y_i \le y_i^+, \tag{7}$$

where $y^+, y^- \ge 0$; $u \in \{0, 1\}$; and M is a large enough positive constant. With this formula, the PFDF term in (1) can be transformed into 2^n new signomial terms with positive variables, as shown in Eq. (8).

$$D(\mathbf{y}) = c \prod_{i=1}^{n} \left((y_i^+)^{\alpha_i} + (-1)^{\alpha_i} (y_i^-)^{\alpha_i} \right). \tag{8}$$

Although all the new variables in (8) are positive, the error of division by zero arises. The following program may be considered instead.

$$\text{Min } \{y_1^{-1} | y_1 \in \{-4, -1, 1, 5\}\}. \tag{9}$$

Let $y_1 = y_1^+ - y_1^-$, where $y_1^+, y_1^- \ge 0$, the above program can be transformed into the following program based on (6) and (7).

$$\begin{aligned}
&\text{Min } (y_1^+)^{-1} - (y_1^-)^{-1} \\
&\text{s.t. } y_1 = -4u_1 - 1u_2 + u_3 + 5u_4, \qquad u_1 + u_2 + u_3 + u_4 = 1, \\
&-y_1^- \le y_1 \le Mu - y_1^-, \qquad y_1^+ + M(u - 1) \le y_1 \le y_1^+,
\end{aligned}$$

where $u, u_1, u_2, u_3,$ and u_4 are binary variables.

The global optimal solution of the above program is unbounded. Since the MINLP solver drives the objective value to be as small as possible, it forces $y_1^+ \to 5$ and $y_1^- \to 0$ so that the objective value is infinitesimal. However, the global optimal solution of the program in (9) is -1 with $y_1 = -1$.

The methods proposed by Floudas et al. and Pörn et al. are applicable in solving optimization problems with continuous variables. However, they are not appropriate for efficiently handling PFDF terms because they lack an effective approach to treat free-sign discrete variables. Recently, Li and Lu [15] introduced a novel method that specifically handles PFDF terms without using the exponential transformation technique. Instead, their method transforms a PFDF term into a series of linear constraints. For example, a simple PFDF term $z_{123} = y_1^{-4/3} y_2^3 y_3^{-2}$, where $y_i \in \{d_{i,1}, d_{i,2}, \ldots, d_{i,r_i}\}$ for $i = 1, 2, 3$, can be transformed into the following linear constraints:

$$y_i = \sum_{k=1}^{r_i} d_{i,k} p_{i,k}, \quad \sum_{k=1}^{r_1} p_{i,k} = 1, \quad i = 1, 2, 3, \tag{10}$$

$$z_1 = y_1^{-\frac{4}{3}} = \sum_{k=1}^{r_1} d_{1,k}^{-\frac{4}{3}} p_{1,k}, \tag{11}$$

$$d_{2,k}^3 z_1 - M_2(p_{2,k} - 1) \le z_{12} = y_1^{-4/3} y_2^3 \le d_{2,k}^3 z_1 + M_2(p_{2,k} - 1) \quad \forall k, \tag{12}$$

$$d_{3,k}^{-2} z_2 - M_3(p_{3,k} - 1) \le z_{123} \le d_{3,k}^{-2} z_2 + M_3(p_{3,k} - 1) \quad \forall k, \tag{13}$$

where $p_{i,k} \in \{0, 1\}$, M_2, and M_3 are large enough positive constants. Li and Lu [15] developed a technique to reduce the number of binary variables required. Their technique replaces the original $\sum_{i=1}^{3} d_{i,r_i}$ binary variables ($p_{1,k}, p_{2,k},$ and $p_{3,k}$) in (10)–(13) with $\sum_{i=1}^{3} \lceil \log_2 d_{i,r_i} \rceil$ binary variables. The smaller number of binary variables improves the efficiency of solving PFDF terms; however, the technique generates numerous overall constraints with the Big-M parameters (M_2 and M_3). Moreover, the efficiency of the formulations in inequalities (12) and (13) is strongly affected by the values of the Big-M parameters.

Remark 2. In linearizing the PFDF term in (1), the methods of Li and Lu [15] require $\sum_{i=1}^{n} \lceil \log_2 r_i \rceil$ binary variables, $1 + \sum_{i=1}^{n} (3 + 4\lceil \log_2 r_i \rceil) + 2\sum_{i=2}^{n} r_i$ linear constraints, and $\sum_{i=1}^{n} (r_i + \lceil \log_2 r_i \rceil)$ auxiliary non-negative continuous variables.

Current deterministic methods are based on the concept of continuous variables; therefore, they always transform the original model into another programming model (e.g., DC programming, convex programming) and solve it with a commercial solver. Due to the different nature of a discrete variable from that of a continuous variable, developing a novel method to address the above problems is necessary. A concise and efficient method that linearizes PFDF term into a set of linear inequalities directly without redundant transformation is proposed, and further, the proposed method only requires logarithmic numbers of binary variables and constraints. Therefore, the original Program 1 is transformed into the following program.

Program 2.

$$\text{Min} \sum_{p=1}^{T} L_p(\mathbf{y}) + \sum_{p=1}^{T'} F_p(\mathbf{x})$$

$$\text{s.t.} \sum_{p=1}^{T_w} L_{w,p}(\mathbf{y}) + \sum_{p=1}^{T'_w} F_{w,p}(\mathbf{x}) \le l_w, \quad w = 1, \ldots, s,$$

where $L_p(\mathbf{y})$ and $L_{w,p}(\mathbf{y})$ are the sets of linear inequalities that replace the original PFDF terms ($D_p(\mathbf{y})$ and $D_{w,p}(\mathbf{y})$) in Program 1. The advantages of the proposed model are as follows:

(i) A logarithmic number of binary variables and constraints are used to solve the PFDF term. For example, in (1), only $\sum_{i=1}^{n}\lceil \log_2 r_i \rceil$ binary variables and $2 + 2\lceil \log_2 r_1 \rceil + \sum_{i=2}^{n}(3 + 7\lceil \log_2 r_i \rceil)$ constraints are required to linearize a PFDF term.

(ii) It can handle free-sign discrete variables directly without any additional transformation.

(iii) Program 2 becomes a Mixed-Integer Linear Programming (MIP) problem if $\sum_{p=1}^{T'} f_p$ and $\sum_{p=1}^{T'_w} f_{w,p}$ are linear functions. Thus, many existing branch-and-cut techniques can be used to improve the computational efficiency of the program.

After submitting this article, the author noticed that Henry [16] proposed a similar formulation for solving the PFDF term in his Ph.D. dissertation. Based on their respective submitted dates, both studies, i.e., the present study and Henry's dissertation [16], are clearly results of independent research.

The remainder of this paper is organized as follows. Section 2 present the proposed modified SOS1 constraint model that can hold a specific value in any one of a series of continuous variables, where the value of every other variable is zero. In Section 3, a novel method for linearizing PFDF terms is proposed, along with an explanation on why the method only needs logarithmic numbers of constraints and binary variables. In Section 4, numerical examples are provided to demonstrate the advantages of the proposed model. Section 5 contains some concluding remarks.

2. The modified SOS1 constraint model

In general, an SOS1 constraint model with size r will require r binary variables. This study, however, proposes a novel SOS1 constraint model with a size r that uses $\lceil \log_2 r \rceil$ binary variables and $2\lceil \log_2 r \rceil$ constraints. This novel SOS1 constraint will construct r auxiliary variables, which have the characteristics of binary variables but are in fact continuous variables. The r auxiliary continuous variables only have two values: zero or a specific positive value. As the specific value can only be assigned to one variable at a time, the value of every other variable is zero. The auxiliary variables are referred to as *two-value continuous variables* with the SOS1 property.

First, a sequence of *two-value continuous variables* with SOS1 constraints (p_k ($k = 1, \ldots, r$)) is constructed. Only one variable in the sequence has the value 1, and the others are zero. For many years, the SOS1 constraint was constructed by a mixed-integer binary model [17,18]. However, Vielma and Nemhauser [19] proposed a method that constructs the SOS1 constraint with a logarithmic number of binary variables and constraints. Their method can be described by the following proposition.

Proposition 1. *Given two indicant sets $K = \{1, \ldots, r\}$ and $J = \{1, \ldots, h = \lceil \log_2 r \rceil\}$, the two-value continuous variables, p_k ($k \in K$), with the SOS1 property can be constructed by adding a logarithmic number of binary variables, u_j ($j \in J$), and the following constraints.*

$$\sum_{k\in K} p_k = 1, \tag{14}$$

$$\sum_{k\in J^+(j)} p_k \le u_j, \qquad \sum_{k\in J^-(j)} p_k \le (1 - u_j) \quad \forall j \in J, \tag{15}$$

where

$$\text{(i) } J^+(j) = \{k \in K : j \in \sigma(B(k))\}, \qquad J^-(j) = \{k \in K : j \notin \sigma(B(k))\}; \tag{16}$$

(ii) $B : K \to \{0, 1\}^J$ *is any injective function; and*

(iii) $\sigma(B(k))$ *is the support of vector* $B(k)$.

Table 1
An instance of the injective function and the relationships between the variables.

k	$B(k)$ $[u_1 \quad u_2 \quad u_3]^T$	$\sigma(B(k))$	Instance 1							Instance 2							
			p_1	p_2	p_3	p_4	p_5	p_6	p_7	q_1	q_2	q_3	q_4	q_5	q_6	q_7	q_8
1	$[0 \quad 0 \quad 0]^T$	$\varnothing$	1	0	0	0	0	0	0	z	0	0	0	0	0	0	0
2	$[1 \quad 0 \quad 0]^T$	$\{1\}$	0	1	0	0	0	0	0	0	z	0	0	0	0	0	0
3	$[0 \quad 1 \quad 0]^T$	$\{2\}$	0	0	1	0	0	0	0	0	0	z	0	0	0	0	0
4	$[1 \quad 1 \quad 0]^T$	$\{1, 2\}$	0	0	0	1	0	0	0	0	0	0	z	0	0	0	0
5	$[0 \quad 0 \quad 1]^T$	$\{3\}$	0	0	0	0	1	0	0	0	0	0	0	z	0	0	0
6	$[1 \quad 0 \quad 1]^T$	$\{1, 3\}$	0	0	0	0	0	1	0	0	0	0	0	0	z	0	0
7	$[0 \quad 1 \quad 1]^T$	$\{2, 3\}$	0	0	0	0	0	0	1	0	0	0	0	0	0	z	0
8	$[1 \quad 1 \quad 1]^T$	$\{1, 2, 3\}$								0	0	0	0	0	0	0	Z

Because $B : K \to \{0, 1\}^J$ is an injective function, some combinations of $\{0, 1\}^J$ will have a lack of disjunctive constraints when $|K|$ is not a power of two ($|K| < 2^J$). Therefore, the constraints in (15) cannot prevent the procedure from executing the combinations of $\{0, 1\}^J$ that do not have a mapping disjunctive constraint. However, if the procedure does execute those combinations of $\{0, 1\}^J$, the solutions will be infeasible because all of the two-value continuous variables are zero.

Proof. Following Vielma and Nemhauser [19], constraints (14) and (15) are used to construct the SOS1 property. □

The following instance illustrates the application of the SOS1 constraints proposed in Proposition 1.

Instance 1. Given two indicant sets $K = \{1, 2, 3, 4, 5, 6, 7\}, J = \{1, 2, 3\}$, and an arbitrary injective function B, every element in the domain K is mapped to one element in the range $\{0, 1\}^3$. Table 1 shows an instance of the injective function B. The relationships between the *two-value continuous variables* p_k $(k \in K)$ and binary variables u_j $(j \in J)$ are expressed as follows:

$$
\begin{aligned}
&p_1 + p_2 + p_3 + p_4 + p_5 + p_6 + p_7 = 1, \\
&p_2 + p_4 + p_6 \le u_1, \qquad p_1 + p_3 + p_5 + p_7 \le 1 - u_1, \\
&p_3 + p_4 + p_7 \le u_2, \qquad p_1 + p_2 + p_5 + p_6 \le 1 - u_2, \\
&p_5 + p_6 + p_7 \le u_3, \qquad p_1 + p_2 + p_3 + p_4 \le 1 - u_3.
\end{aligned}
$$

Although the above formulation lacks disjunctive constraints for the combination of $k = 8$ $(u_1 = 1, u_2 = 1, u_3 = 1)$, the procedure cannot execute the combination of $k = 8$ successfully. If it does execute the combination, it will yield an infeasible solution $p_1 = p_2 = p_3 = p_4 = p_5 = p_6 = p_7 = 0$.

Proposition 2 proposes a sequence of *two-value continuous variables* based on Proposition 1.

Proposition 2. *z is a positive variable. For a sequence of non-negative continuous variables q_k $(k \in K)$, if*

$$\sum_{k \in K} q_k = z, \tag{17}$$

$$\sum_{k \in J^+(j)} q_k \le v_j, \qquad \sum_{k \in J^-(j)} q_k \le z - v_j \quad \forall j \in J, \tag{18}$$

$$z - M(1 - u_j) \le v_j \le z + M(1 - u_j) \quad \forall j \in J, \tag{19}$$

$$v_j \le M u_j \quad \forall j \in J, \tag{20}$$

where $J^+(j)$ and $J^-(j)$ are the same as in (16) for Proposition 1, u_j represents the binary variables, and $M = \max\{z\}$ is a large enough constant. Then, only one of q_k can be assigned a specific value by variable z, and the value of every other variable is zero in any combination of $\{0, 1\}^J$. The non-negative continuous variables then become the two-value continuous variables.

Proof. If $u_j = 1$, it will force $v_j = z$ by constraint (19). Otherwise, $v_j = 0$ by constraint (20). Therefore, the variables v_j are equal to the product terms zu_j, and inequality (18) can be rewritten as inequality (25) as follows:

$$\sum_{k \in J^+(j)} q_k \le z u_j, \qquad \sum_{k \in J^-(j)} q_k \le z(1 - u_j) \quad \forall j \in J. \tag{21}$$

Constraints (17) and (21) are similar to constraints (14) and (15). Hence, (17) and (21) will force $q_k = 0$ for all $[u_{,} \ldots, u_j]^T \neq B(k)$. □

Instance 2 illustrates the *two-value continuous variables* proposed in Proposition 2.

Instance 2. Given two indicant sets $K = \{1, 2, 3, 4, 5, 6, 7, 8\}$ and $J = \{1, 2, 3\}$, as well as the injective function B described in Instance 1, the *two-value continuous variables*, q_k $(k \in K)$, based on Proposition 2, can be constructed by the following constraints.

$$\begin{aligned}
&q_1 + q_2 + q_3 + q_4 + q_5 + q_6 + q_7 + q_8 = z,\\
&q_2 + q_4 + q_6 + q_8 \le v_1, \qquad q_1 + q_3 + q_5 + q_7 \le z - v_1,\\
&q_3 + q_4 + q_7 + q_8 \le v_2, \qquad q_1 + q_2 + q_5 + q_6 \le z - v_2,\\
&q_5 + q_6 + q_7 + q_8 \le v_3, \qquad q_1 + q_2 + q_3 + q_4 \le z - v_3,\\
&z - M(1 - u_j) \le v_j \le z + M(1 - u_j) \quad \forall j \in J,\\
&v_j \le Mu_j \quad \forall j \in J,
\end{aligned}$$

where $M = \max\{z\}$ is a big enough constant. The relationships between the *two-value continuous variables* q_k $(k \in K)$ and binary variables u_j $(j \in J)$ are listed in Table 1. The value of variable z can be any positive real number.

3. Proposed logarithmic method

A discrete nonlinear function $z = c\prod_{i=1}^{n} g_i(y_i)$, where $y_i \in \{d_{i,1}, d_{i,1}, \ldots, d_{i,r_i}\}$, contains $\prod_{i=1}^{n} r_i$ combinations. Current methods need at least $\sum_{i=1}^{n} r_i$ constraints and numerous binary variables to convert this function into a linear programming or convex programming problem. In this section, a novel method that only needs a logarithmic number of constraints and binary variables to linearize the discrete nonlinear function is presented.

Proposition 3. *Given a nonlinear function $g_1(y_1)$, where y_1 is a free-sign discrete variable, $K_1 = \{1, \ldots, r_1\}$ can be an indicant set of the values $d_{1,k}$ in variable y_1. Thus, $y_1 \in \{d_{1,1}, d_{1,2}, \ldots, d_{1,r_1}\}$ and $d_{1,k}$ are free-sign values for $k \in K_1$. In addition, $J_1 = \{1, \ldots, h = \lceil \log_2 r_1 \rceil\}$ may be an indicant set of binary variables $u_{1,j}$. Then, $g_1(y_1)$ can be expressed as follows:*

$$g_1(y_1) = \sum_{k \in K_1} g_1(d_{1,k})p_{1,k},$$

$$\sum_{k \in K_1} p_{1,k} = 1, \tag{22}$$

$$\sum_{k \in J_1^+(j)} p_{1,k} \le u_{1,j}, \qquad \sum_{k \in J_1^-(j)} p_{1,k} \le (1 - u_{1,j}) \quad \forall j \in J_1, \tag{23}$$

where $J_1^+(j) = \{k \in K_1 : j \in \sigma(B(k))\}$ and $J_1^-(j) = \{k \in K_1 : j \notin \sigma(B(k))\}$ are the same as (16) in Proposition 1.

Proof. Based on Proposition 1, constraints (22)–(23) are used to construct the *two-value continuous variables* $p_{1,k}$ with the SOS1 property. Thus, $\sum_{k \in K_1} g_1(d_{1,k})p_{1,k}$ can be any one of the values $\{g_1(d_{1,1}), \ldots, g_1(d_{1,r_1})\}$; therefore, $\sum_{k \in K_1} g_1(d_{1,k})p_{1,k} = g_1(y_1)$. □

Remark 3. Only $\lceil \log_2 r_1 \rceil$ binary variables, $2 + 2\lceil \log_2 r_1 \rceil$ linear constraints, and r_1 auxiliary non-negative continuous variables are required in Proposition 3.

Proposition 4. *Given a product term $z_2 = g_1(y_1)g_2(y_2)$, $g_i(y_i)$ are nonlinear functions of free-sign discrete variables y_i for $i = 1, 2$. $y_i \in \{d_{i,1}, d_{i,2}, \ldots, d_{i,r_i}\}$, and all $d_{i,k}$ can be free-sign values. In addition, $K_i = \{1, \ldots, r_i\}$ can be the indicant sets of all possible values $d_{1,k}$ in variables y_i; and $J_i = \{1, 2, \ldots, h_i = \lceil \log_2 r_i \rceil\}$ represents the indicant sets of binary variables $u_{i,j}$ for $i = 1, 2$. Denote $z_1 = g_1(y_1)$, then z_2 can be expressed as follows:*

$$z_1 = \sum_{k \in K_1} g_1(d_{1,k})p_{1,k}, \tag{24}$$

$$z_2 = \sum_{k \in K_2} (g_2(d_{2,k})(q_{2,k} + p_{2,k} \cdot m_1)), \tag{25}$$

$$\sum_{k \in K_i} p_{i,k} = 1, \quad i = 1, 2, \tag{26}$$

$$\sum_{k \in J_i^+(j)} p_{i,k} \le u_{i,j}, \qquad \sum_{k \in J_i^-(j)} p_{i,k} \le (1 - u_{i,j}) \quad \forall j \in J_i,\ i = 1, 2, \tag{27}$$

$$\sum_{k \in K_2} q_{2,k} = z_1 - m_1, \tag{28}$$

$$\sum_{k\in J_2^+(j)} q_{2,k} \le v_{2,j}, \qquad \sum_{k\in J_2^-(j)} q_{2,k} \le (z_1 - m_1 - v_{2,j}) \quad \forall j \in J_2, \tag{29}$$

$$z_1 - m_1 - M_1(1-u_{2,j}) \le v_{2,j} \le z_1 - m_1 + M_1(1-u_{2,j}) \quad \forall j \in J_2, \tag{30}$$

$$v_{2,j} \le M_1 u_{2,j} \quad \forall j \in J_2, \tag{31}$$

where $u_{i,j} \in \{0,1\}\ \forall j \in J_i,\ i = 1,2,\ m_1 = \min\{0, z_1\}$, *and* $M_1 = \max\{z_1\} - m_1$.

Proof. (i) Similar to Proposition 3, constraints (26)–(27) are used to construct the variables $p_{1,k}$ and $p_{2,k}$ with the SOS1 property. Then, $\sum_{k\in K_1} g_1(d_{1,k})p_{1,k} = g_1(y_1)$ and $\sum_{k\in K_2} g_2(d_{2,k})p_{2,k} = g_2(y_2)$.

(ii) $z_2 = g_2(y_2)z_1 = \sum_{k\in K_2}[g_2(d_{2,k})p_{2,k}z_1] = \sum_{k\in K_2} g_2(d_{2,k})[p_{2,k}(z_1 - m_1 + m_1)]$

$$z_2 = \sum_{k\in K_2} g_2(d_{2,k})[p_{2,k}(z_1 - m_1) + (p_{2,k}m_1)]. \tag{32}$$

(iii) Based on Proposition 2, $q_{2,k}$ are two-value continuous variables by constraints (28)–(31). The values of $q_{2,k}$ are either zero or $(z_1 - m)$, depending on the value of $p_{2,k}$; therefore, Eq. (32) with $q_{2,k} = p_{2,k}(z_1 - m_1)$ can be rewritten as Eq. (25). □

In the following instance, a simple PFDF term with two free-sign discrete variables is used to explain the method proposed in Proposition 4.

Instance 3. Given a PFDF term $z_2 = y_1^3 \cdot y_2^2$, where $y_1 \in \{d_{1,1}, \ldots, d_{1,7}\} = \{-3.5, -1, -0.7, 0, 3, 5, 7\}$ and $y_2 \in \{d_{2,1}, \ldots, d_{2,8}\} = \{-9, -7, -4, -1, 1, 3, 4, 5\}$, $K_1 = \{1, 2, \ldots, 7\}$ and $K_2 = \{1, 2, \ldots, 8\}$ represent the indicant sets of discrete values in variables y_1 and y_2, respectively. The instance of the injective function B is presented in Table 1. Denote $z_1 = y_1^3$ and $z_2 = z_1 \cdot y_2^2$; then z_1 and z_2 can be expressed as follows:

$$\begin{aligned}
& z_1 = \sum_{k\in K_1} d_{1,k}^3 p_{1,k}, \quad \sum_{k\in K_1} p_{1,k} = 1 \\
& p_{1,2} + p_{1,4} + p_{1,6} \le u_{1,1}, \qquad p_{1,1} + p_{1,3} + p_{1,5} + p_{1,7} \le 1 - u_{1,1}, \\
& p_{1,3} + p_{1,4} + p_{1,7} \le u_{1,2}, \qquad p_{1,1} + p_{1,2} + p_{1,5} + p_{1,6} \le 1 - u_{1,2}, \\
& p_{1,5} + p_{1,6} + p_{1,7} \le u_{1,3}, \qquad p_{1,1} + p_{1,2} + p_{1,3} + p_{1,4} \le 1 - u_{1,3}, \\
& z_2 = \sum_{k\in K_2} d_{2,k}^2 (q_{2,k} + p_{2,k} \cdot m_1), \quad \sum_{k\in K_2} p_{2,k} = 1 \\
& p_{2,2} + p_{2,4} + p_{2,6} + p_{2,8} \le u_{2,1}, \qquad p_{2,1} + p_{2,3} + p_{2,5} + p_{2,7} \le 1 - u_{2,1}, \\
& p_{2,3} + p_{2,4} + p_{2,7} + p_{2,8} \le u_{2,2}, \qquad p_{2,1} + p_{2,2} + p_{2,5} + p_{2,6} \le 1 - u_{2,2}, \\
& p_{2,5} + p_{2,6} + p_{2,7} + p_{2,8} \le u_{2,3}, \qquad p_{2,1} + p_{2,2} + p_{2,3} + p_{2,4} \le 1 - u_{2,3}, \\
& \sum_{k\in K_2} q_{2,k} = z_1 - m_1, \\
& z_1 - m_1 - M_1(1-u_{2,j}) \le v_{2,j} \le z_1 - m_1 + M_1(1-u_{2,j}), \quad j = 1,2,3, \\
& v_{2,j} \le M_1 u_{2,j}, \quad j = 1,2,3, \\
& q_{2,2} + q_{2,4} + q_{2,6} + q_{2,8} \le v_{2,1}, \qquad q_{2,1} + q_{2,3} + q_{2,5} + q_{2,7} \le z_1 - m_1 - v_{2,1}, \\
& q_{2,3} + q_{2,4} + q_{2,7} + q_{2,8} \le v_{2,2}, \qquad q_{2,1} + q_{2,2} + q_{2,5} + q_{2,6} \le z_1 - m_1 - v_{2,2}, \\
& q_{2,5} + q_{2,6} + q_{2,7} + q_{2,8} \le v_{2,3}, \qquad q_{2,1} + q_{2,2} + q_{2,3} + q_{2,4} \le z_1 - m_1 - v_{2,3},
\end{aligned}$$

where $u_{i,1}, u_{i,2}$, and $u_{i,3}$ are binary variables for $i = 1, 2$; $m_1 = \min\{0, z_1\} = (-3.5)^3 = -42.875$; and $M_1 = \max\{z_1\} - m_1 = 7^3 + 42.875 = 385.875$.

The main result can be deduced as follows.

Theorem 1. *Denote* $z_1 = g_1(y_1)$ *and* $z_n = z_{n-1} \cdot g_n(y_n) = \prod_{i=1}^n g_i(y_i)$, *where* y_i *are free-sign discrete variables and* $y_i \in \{d_{i,1}, d_{i,2}, \ldots, d_{i,r_i}\}$ *for* $i = 1, \ldots, n$. $K_i = \{1, \ldots, r_i\}$ *represents the indicant sets of values* $d_{i,k}$ *in* y_i *for* $i = 1, \ldots, n$; *and* $J_i = \{1, 2, \ldots, h_i = \lceil \log_2 r_i \rceil\}$ *represents the indicant sets of binary variables* $u_{i,j}$ *for* $i = 1, \ldots, n$. *The term* z_n *can be expressed as the following linear constraints:*

$$\begin{aligned}
& y_i = \sum_{k\in K_i} d_{i,k} p_{i,k}, \quad i = 1, \ldots, n, \\
& (24), \\
& (26), (27), \quad i = 1, \ldots, n, \\
& z_i = \sum_{k\in K_i} (g_i(d_{i,k})(q_{i,k} + p_{i,k} \cdot m_{i-1})), \quad i = 2, \ldots, n,
\end{aligned}$$

Table 2
Comparison of two methods for handling the PFDF term in (1).

Item	Method			
	Floudas [8–11]	Pörn et al.'s [12]	Li and Lu [15]	Proposed method
No. of binary variables	$\sum_{i=1}^{n}(r_i-1)$	$\sum_{i=1}^{n}(r_i-1)$	$\sum_{i=1}^{n}\lceil\log_2 r_i\rceil$	$\sum_{i=1}^{n}\lceil\log_2 r_i\rceil$
No. of extra linear constraints	$2n$	$2n$	$\sum_{i=1}^{n}(3+4\lceil\log_2 r_i\rceil)+2\sum_{i=2}^{n} r_i+1$	$3+2\lceil\log_2 r_1\rceil+\sum_{i=2}^{n}(4+7\lceil\log_2 r_i\rceil)$
No. of extra auxiliary continuous variables	n	n	$\sum_{i=1}^{n}(r_i+\lceil\log_2 r_i\rceil)$	$r_1+\sum_{i=2}^{n}(2r_i+\lceil\log_2 r_i\rceil)$
Handle free-sign variable	No	Yes, it will produce $\prod_{i=1}^{n}(\alpha_i+1)-1$ new PFDF terms	*Yes*	*Yes*

$$\sum_{k\in K_i} q_{i,k} = z_{i-1} - m_{i-1}, \quad i = 2, \dots, n,$$

$$\sum_{k\in J_i^+(j)} q_{i,k} \le v_{i,j}, \qquad \sum_{k\in J_i^-(j)} q_{i,k} \le z_{i-1} - m_{i-1} - v_{i,j}, \quad \forall j \in J_i,\ i = 2, \dots, n,$$

$$z_{i-1} - m_{i-1} - M_{i-1}(1-u_{i,j}) \le v_{i,j} \le z_{i-1} - m_{i-1} + M_{i-1}(1-u_{i,j}), \quad \forall j \in J_i,\ i = 2, \dots, n,$$

$$v_{i,j} \le M_{i-1}u_{i,j} \quad \forall j \in J_i,\ i = 2, \dots, n,$$

where $m_i = \min\{0, z_i\}$, $M_i = \max\{z_i\} - m_i$, *for* $i = 1, \dots, n-1$.

Proof. The proof is similar to that of Proposition 4. □

Remark 4. Only $\sum_{i=1}^{n}\lceil\log_2 r_i\rceil$ binary variables, $(2+2\lceil\log_2 r_1\rceil)+\sum_{i=2}^{n}(3+7\lceil\log_2 r_i\rceil)$ linear constraints, and $r_1+\sum_{i=2}^{n}(2r_i+\lceil\log_2 r_i\rceil)$ auxiliary non-negative continuous variables are used in Theorem 1.

Table 2 compares the current methods and proposed method for linearizing the PFDF term in (1). In the next section, some numerical examples are used to compare the computational efficiency of the methods.

4. Numerical examples

To demonstrate the efficiency of the proposed model, five numerical examples solved on a PC with a 3.16 GHz Intel Core™ 2 Duo CPU and 4 GB RAM are presented.

Example 1. In this example, only one PFDF term is in the objective function; the other terms are linear constraints.

$$\text{Max/Min } y_1^{-\frac{4}{3}} y_2^3 y_3^{-2}$$

$$\text{s.t. } y_1 + y_2 + y_3 \le 10, \tag{33}$$

$$y_1 + y_2 + y_3 \ge -4, \tag{34}$$

$$0 < y_1 \le 4, \qquad -4 \le y_2 \le 3, \qquad -4 \le y_3 \le 4, \tag{35}$$

where y_1 is a positive discrete variable, and y_2 and y_3 are free-sign discrete variables. To compare the computational efficiency of Li and Lu's method and the proposed method, y_1, y_2, and y_3 have different r discrete values ($r = 8, 128, 256$, or 512) in this example. Then, $y_1 \in \{d_{1,1}, d_{1,2}, \dots, d_{1,r}\} = \{\frac{4*1}{r}, \frac{4*2}{r}, \dots, \frac{4*r}{r}\}$, $y_2 \in \{d_{2,1}, d_{2,2}, d_{2,3}, \dots, d_{2,r}\} = \{-4, -4+\frac{7*1}{r-1}, -4+\frac{7*2}{r-1}, \dots, 3\}$, and $y_3 \in= \{d_{3,1}, d_{3,2}, d_{3,3}, \dots, d_{3,r}\} = \{-4, -4+\frac{8*1}{r-1}, -4+\frac{8*2}{r-1}, \dots, 4\}$.

Denote $z_1 = y_1^{-4/3}$, $z_2 = y_1^{-4/3}y_2^3 = y_2^3 \cdot z_1$, and $z_3 = y_1^{-4/3}y_2^3y_3^{-2} = y_3^{-2} \cdot z_2$. Li and Lu's method requires $3\lceil\log_2 r\rceil$ binary variables and $10 + 12\lceil\log_2 r\rceil + 4r$ additional linear constraints to linearize the PFDF term z_3. Under their method, the problem can be reformulated as follows.

Program 3.

$$\text{Max/Min } z_3$$

s.t. (10)–(13)

(33)–(35).

In contrast, the proposed method (i.e., Theorem 1) only requires $3\lceil\log_2 r\rceil$ binary variables and $11+16\lceil\log_2 r\rceil$ additional linear constraints to linearize the z_3 term. $K_i = \{1, 2, \dots, r_i\}$ denotes the indicant sets of all possible values $d_{i,k}$ in y_i for $i = 1, 2, 3$; and $J_i = \{1, 2, \dots, h_i = \lceil\log_2 r_i\rceil\}$ denotes the indicant sets of binary variables $u_{i,j}$ for $i = 1, 2, 3$. The injective function B is similar to that in Table 1. Based on Theorem 1, this problem can be reformulated as follows. Here, the program is only shown with $r = 8$.

Program 4.

$$
\begin{aligned}
&\text{Max/Min } z_3\\
&\text{s.t. } y_1=\sum_{k\in K_1} d_{1,k}p_{1,k},\qquad z_1=\sum_{k\in K_1} d_{1,k}^{-\frac{4}{3}}p_{1,k},\quad \sum_{k\in K_1} p_{1,k}=1,\\
&y_2=\sum_{k\in K_2} d_{2,k}p_{2,k},\qquad z_2=\sum_{k\in K_2} d_{2,k}^{3}(q_{2,k}+p_{2,k}\cdot m_1),\quad \sum_{k\in K_2} p_{2,k}=1,\ \sum_{k\in K_2} q_{2,k}=z_1-m_1,\\
&y_3=\sum_{k\in K_3} d_{3,k}p_{3,k},\qquad z_3=\sum_{k\in K_3} d_{3,k}^{-2}(q_{3,k}+p_{3,k}\cdot m_2),\quad \sum_{k\in K_3} p_{3,k}=1,\ \sum_{k\in K_3} q_{3,k}=z_2-m_2,\\
&p_{i,2}+p_{i,4}+p_{i,6}+p_{i,8}\le u_{i,1},\qquad p_{i,1}+p_{i,3}+p_{i,5}+p_{i,7}\le 1-u_{i,1},\quad i=1,2,3,\\
&p_{i,3}+p_{i,4}+p_{i,7}+p_{i,8}\le u_{i,2},\qquad p_{i,1}+p_{i,2}+p_{i,5}+p_{i,6}\le 1-u_{i,2},\quad i=1,2,3,\\
&p_{i,5}+p_{i,6}+p_{i,7}+p_{i,8}\le u_{i,3},\qquad p_{i,1}+p_{i,2}+p_{i,3}+p_{i,4}\le 1-u_{i,3},\quad i=1,2,3,\\
&z_{i-1}-m_{i-1}-M_{i-1}(1-u_{i,j})\le v_{i,j}\le z_{i-1}-m_{i-1}+M_{i-1}(1-u_{i,j}),\quad j=1,2,3,\ i=2,3,\\
&v_{i,j}\le M_{i-1}u_{i,j},\quad j=1,2,3,\ i=2,3,\\
&q_{i,2}+q_{i,4}+q_{i,6}+q_{i,8}\le v_{i,1},\qquad q_{i,1}+q_{i,3}+q_{i,5}+q_{i,7}\le z_{i-1}-m_{i-1}-v_{i,1},\quad i=2,3,\\
&q_{i,3}+q_{i,4}+q_{i,7}+q_{i,8}\le v_{i,2},\qquad q_{i,1}+q_{i,2}+q_{i,5}+q_{i,6}\le z_{i-1}-m_{i-1}-v_{i,2},\quad i=2,3,\\
&q_{i,5}+q_{i,6}+q_{i,7}+q_{i,8}\le v_{i,3},\qquad q_{i,1}+q_{i,2}+q_{i,3}+q_{i,4}\le z_{i-1}-m_{i-1}-v_{i,3},\quad i=2,3,\\
&(33)\text{–}(35),
\end{aligned}
$$

where $u_{i,1}$, $u_{i,2}$, and $u_{i,3}$ are binary variables for $i = 1, 2, 3$; $m_1 = 0$, $m_2 \approx -161.27$, $M_1 \approx 2.5199$, and $M_2 \approx 229.307$.

Both of the compared methods can reformulate a PFDF term as a set of linear constraints. Thus, this example can be converted into an MIP problem that can be solved using ILOG CPLEX 11 [20] as the MIP solver in both methods. The related solutions, CPU time, number of simplex iterations, number of binary variables, and number of constraints are listed in Table 3. All the values of r demonstrate that the proposed method is much more efficient than Li and Lu's method, especially when r is large.

Example 2. In this example, taken from Li and Lu [15], has several signomial terms in the objective function and constraints. To demonstrate the efficiency of the proposed method, all signomial terms are transformed into PFDF terms by converting variables x_1 and x_2 into discrete variables. The problem is formulated as follows:

$$
\begin{aligned}
&\text{Min } x_1^3x_2y_1^3y_2+x_1^3x_2y_1y_2^2\\
&\text{s.t. } x_1^3x_2y_1^2+y_1y_2\le -500,\\
&-x_1^3x_2y_1+y_1^2y_2\le 500,\\
&-6\le x_1\le 6.75,\qquad -6\le x_2\le 6.75,\qquad -1\le y_1\le 9.2,\qquad -9\le y_1\le 6.3,
\end{aligned}
$$

where x_1, x_2, x_3, and x_4 are coded with free-sign integer multiples of 0.05, 0.05, 0.04, and 0.06, respectively.

To achieve global optimization, the six PFDF terms $x_1^3x_2y_1^3y_2$, $x_1^3x_2y_1y_2^2$, $x_1^3x_2y_1^2$, y_1y_2, $x_1^3x_2y_1$, and $y_1^2y_2$ need to be linearized as linear constraints. Denote $z_x = x_1^3x_2$. Based on Theorem 1, 16 binary variables and 79 linear constraints are used to linearize the term z_x. Then, the problem is reformulated as follows.

$$
\begin{aligned}
&\text{Min } z_xy_1^3y_2+z_xy_1y_2^2\\
&\text{s.t. } z_xy_1^2+y_1y_2\le -500,\\
&-z_xy_1+y_1^2y_2\le 500,\\
&\text{additional variables and linear constraints for linearizing } z_x.
\end{aligned}
$$

Denote $z_{x1\_32} = z_xy_1^3y_2$, $z_{x12\_2} = z_xy_1y_2^2$, $z_{x1\_2} = z_xy_1^2$, $z_{12} = y_1y_2$, $z_{x1} = z_xy_1$, and $z_{1\_22} = y_1^2y_2$. In linearizing the six PFDF terms based on Theorem 1, then the formulation of this problem is shown as follows.

$$
\begin{aligned}
&\text{Min } z_{x1\_32}+z_{x12\_2}\\
&\text{s.t. } z_{x1\_2}+z_{12}\le -500,\\
&-z_{x1}+z_{1\_22}\le 500,\\
&\text{additional variables and linear constraints for linearizing } z_x,\ z_{x1\_32},\ z_{x12\_2},\ z_{x1\_2},\ z_{12},\ z_{x1},\ \text{and } z_{1\_22}.
\end{aligned}
$$

To solve this problem, ILOG CPLEX 11 [20] is used as the MIP solver for both of the compared methods. The computation results shown in Table 4 demonstrate that the proposed method significantly outperforms Li and Lu's method.

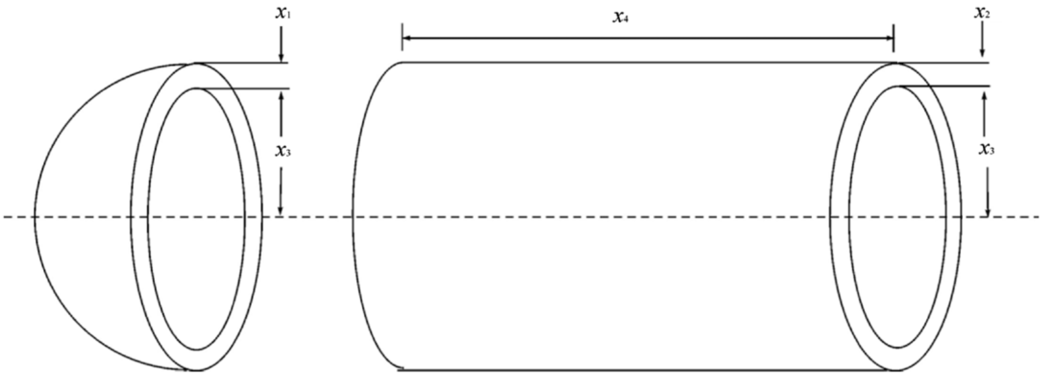

Fig. 1. Pressure vessel design problem.

Table 3
Computation results for Example 1.

r	Item	Maximum		Minimum	
		Li and Lu's method (P3)	Proposed method (P4)	Li and Lu's method (P3)	Proposed method (P4)
8	CPU time (s)	0	0	0	0
	No. of simplex iterations	159	47	167	4
	No. of 0–1 variables	9	9	9	9
	No. of constraints	78	59	78	59
	Solution (y_1, y_2, y_3)	(0.5, 3, 0.57142857)		(0.5,−4, 0.57142857)	
	Objective value	208.359		−493.889	
128	CPU time (s)	1.34	0.14	0.72	0.05
	No. of simplex iterations	32 649	443	13 388	100
	No. of 0–1 variables	21	21	21	21
	No. of constraints	606	123	606	123
	Solution (y_1, y_2, y_3)	(0.03125, 3, 0.03149606)		(0.03125,−4, 0.03149606)	
	Objective value	2 765 144.689		−6 554 417.041	
256	CPU time (s)	1.69	0.17	1.25	0.06
	No. of simplex iterations	22 069	1123	14 015	151
	No. of 0–1 variables	24	24	24	24
	No. of constraints	1130	139	1130	139
	Solution (y_1, y_2, y_3)	(0.015625, 3,−0.01568627)		(0.015625,−4, 0.01568627)	
	Objective value	28 090 800		−66 585 600	
512	CPU time (s)	215.22	1.53	25.31	0.08
	No. of simplex iterations	2 039 007	6509	108 721	185
	No. of 0–1 variables	27	27	27	27
	No. of constraints	2166	155	2166	155
	Solution (y_1, y_2, y_3)	(0.0078125, 3, 0.00782078)		(0.0078125,−4, 0.00782778)	
	Objective value	284 248 953.622		−673 775 297.474	

Table 4
Computation results for Example 2 using CPLEX 11.

Item	Li and Lu's method	Proposed method
CPU time (s)	220. 44	65.22
No. of simplex iterations	1 237 367	652 412
No. of 0–1 variables	32	32
No. of constraints	4236	326
Solution (x_1, x_2, y_1, y_2)	(2.15, −4.5, 6.04 6.3)	
Objective value	−72 805.201	

Example 3. In this example, the pressure vessel design problem, proposed by Sandgren [21], is formulated as an MINLP problem. As shown in Fig. 1, the problem has four variables: x_1 (the thickness of the spherical head), x_2 (the thickness of the shell), x_3 (the radius of the shell), and x_4 (the length of the shell). The objective is to minimize the total cost of materials as well as that of forming and welding the pressure vessel. To demonstrate the efficiency of the proposed method, all variables are converted into discrete variables, and the problem is formulated as follows:

$$\text{Min } 0.6224x_1x_3x_4 + 1.7781x_2x_3^2 + 3.1661x_1^2x_4 + 19.84x_1^2x_3$$
$$\text{s.t. } 0.0193x_3 - x_1 \le 0,$$

Table 5
Computation results for Example 3 using CPLEX 11.

Item	Li and Lu's method	Proposed method
CPU time (s)	71.61	1
No. of simplex iterations	893 713	20 090
No. of 0–1 variables	30	30
No. of constraints	1690	261
Solution (x_1, x_2, x_3, x_4)	(0.8125, 0.4375, 42, 178)	
Objective value	6074.99836016	

Table 6
Computation results for Example 3 using LINGO 11.

Item	Li and Lu's method	Proposed method
CPU time (s)	1355	23
No. of simplex iterations	9 618 052	188 065
Solution (x_1, x_2, x_3, x_4)	(0.8125, 0.4375, 42, 178)	
Objective value	6074.998	

$$
\begin{aligned}
&0.00954x_3 - x_2 \le 0,\\
&1296\,000 - \pi x_3^2 x_4 - \frac{4}{3}\pi x_3^3 \le 0,\\
&x_4 - 240 \le 0,\\
&0.0625 \le x_1 \le 6.1875, \qquad 0.0625 \le x_2 \le 6.1875, \qquad 10 \le x_3 \le 200, \qquad 10 \le x_4 \le 200,
\end{aligned}
\tag{36}
$$

where x_1 and x_2 are discrete variables with discreteness 0.0625, and x_3 and x_4 are integer variables.

Two experiments are conducted to compare the efficiency of Li and Lu's method and the proposed method. The first experiment compares the computational efficiency of the methods using an MIP solver. Under both methods, this problem becomes an MIP problem because the six PFDF terms ($0.6224x_1x_3x_4$, $1.7781x_2x_3^2$, $3.1661x_1^2x_4$, $19.84x_1^2x_3$, $-\pi x_3^2x_4$, and $-4/3\pi x_3^3$) can be reformulated as linear constraints. The computation results solved by ILOG CPLEX 11 [20] are shown in Table 5.

In the second experiment, LINGO 11 [22] is used as an MINLP solver to compare the computational efficiency of Li and Lu's method [15] and the proposed method. The computational results are shown in Table 6. The results in Tables 5 and 6 demonstrate that the proposed method outperforms the compared methods for both the MIP solver and the MINLP solver.

Example 4. This MINLP problem is modified from Tian et al. [23]. The nonlinear functions in this problem contain signomial terms as well as trigonometric functions, exponential functions, and other nonlinear functions. The example demonstrates that the proposed method can solve this nonlinear discrete problem effectively. The problem is formulated as follows:

$$
\begin{aligned}
&\text{Min } (x_1-3)^2\cos(\pi x_1) + (x_2-6)\sin(0.25\pi x_2) + (x_3-2.5)^2(x_2+2)^{-1} + (x_3+2)^3 e^{-x_4}\\
&\text{s.t. } -10 \le x_i \le 10, \quad i = 1, 3, 4; \; -1 \le x_2 \le 19,
\end{aligned}
\tag{37}
$$

where all variables are free-sign discrete variables with discreteness $d (d = 0.1, 0.05)$.

To demonstrate solving the problem with Theorem 1, the original formulation is reformulated as follows.

Program 5.

$$
\begin{aligned}
&\text{Min } g_{11}(x_1)g_{12}(x_1) + g_{21}(x_2)g_{22}(x_2) + g_{31}(x_3)g_{23}(x_2) + g_{32}(x_3)g_4(x_4)\\
&\text{s.t. } (37),
\end{aligned}
$$

where $g_{11}(x_1) = (x_1-3)^2$, $g_{12}(x_1) = \cos(\pi x_1)$, $g_{21}(x_2) = (x_2-6)$, $g_{22}(x_2) = \sin(0.25\pi x_2)$, $g_{31}(x_3) = (x_3-2.5)^2$, $g_{23}(x_{23}) = (x_2+2)^{-1}$, $g_{32}(x_3) = (x_3+2)^3$, and $g_4(x_4) = e^{-x_4}$.

Then, CPLEX 11 [20] is used as the MIP solver to solve this problem with different d values based on the proposed method and Li and Lu's method. The computational results, shown in Table 7, demonstrate that the proposed method is more efficient computationally than Li and Lu's method for solving this MINLP problem.

Example 5. This compression spring design problem example, investigated by Sandgren [21], is a real-world optimization problem involving discrete and integer variables. The problem has three variables: the wire diameter x_1 (discrete variable),

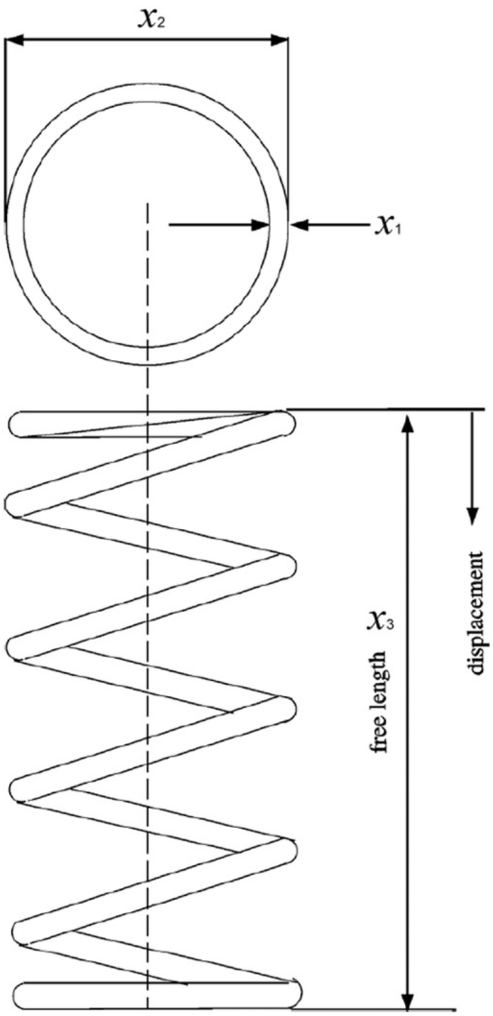

Fig. 2. Compression spring design problem.

the outside diameter x_2 (discrete variable), and the number of active coils x_3(integer variable), as shown in Fig. 2. It is aimed at minimizing the volume of a compression spring under static loading. The original problem is formulated as follows:

$$
\begin{aligned}
&\text{Min } \frac{\pi^2 x_1^2 x_2 (x_3+2)}{4} \\
&\text{s.t. } g_1(\mathbf{x}) = \frac{8 C_f F_{\max} x_2}{\pi x_1^3} - S \le 0, \\
&g_2(\mathbf{x}) = l_f - l_{\max} \le 0, \\
&g_3(\mathbf{x}) = d_{\min} - x_1 \le 0, \qquad (38) \\
&g_4(\mathbf{x}) = x_2 - D_{\max} \le 0, \qquad (39) \\
&g_5(\mathbf{x}) = 3 - \frac{x_2}{x_1} \le 0, \\
&g_6(\mathbf{x}) = \sigma_p - \sigma_{pm} \le 0, \\
&g_7(\mathbf{x}) = \sigma_p + \frac{F_{\max} - F_p}{K} + 1.05(x_3+2)x_1 - l_f \le 0, \\
&g_8(\mathbf{x}) = \sigma_w - \frac{F_{\max} - F_p}{K} \le 0, \\
&C_f = \frac{4(x_2/x_1) - 1}{4(x_2/x_1) - 4} + \frac{0.615 x_1}{x_2},
\end{aligned}
$$

Table 7
Computational results for Example 4.

d values	Item	Li and Lu's method	Proposed method
0.1	CPU time (s)	1.16	0.05
	No. of simplex iterations	4971	304
	No. of 0–1 variables	32	32
	No. of constraints	1752	244
	Solution (x_1, x_2, x_3, x_4)	(−9, 14.9, −10, −10)	
	Objective value	−11 277 692.009	
0.05	CPU time (s)	3.84	0.09
	No. of simplex iterations	6793	545
	No. of 0–1 variables	36	36
	No. of constraints	3368	272
	Solution (x_1, x_2, x_3, x_4)	(−9, 14.85, −10, −10)	
	Objective value	−11 277 692.164	

$$l_f = \frac{F_{\max}}{K} + 1.05(x_3 + 2)x_1,$$
$$K = \frac{Gx_1^4}{8x_3x_2^3},$$
$$\sigma_p = \frac{F_p}{K},$$
$$0.009 \le x_1 \le 0.5, \qquad 0.6 \le x_2 \le 4, \qquad 1 \le x_3 \le 120,$$

where the maximum work load $F_{\max} = 1000$ lb, the maximum shear stress $S = 189\,000.0$ psi, the maximum free length $l_{\max} = 14$ in., the minimum wire diameter $d_{\min} = 0.009$ in., the maximum outside diameter of the spring $D_{\max} = 4$ in., the preload compression force $F_p = 300$ lp, the allowable maximum deflection under preload $\sigma_{pm} = 6$ in., the deflection from preload position to maximum load position $\sigma_w = 1.25$ in., and the shear modulus of the material $G = 11.5 \times 10^6$ psi.

As the original problem is not a standard PFDF form, all of the constraints are rearranged as follows.

$$C_f = \frac{4x_2 - x_1}{4(x_2 - x_1)} + 0.615x_2^{-1}x_1 = (x_2 - 0.25x_1)(x_2 - x_1)^{-1} + 0.615x_1x_2^{-1},$$
$$l_f = F_{\max}K^{-1} + 1.05x_1(x_3 + 2) = 8G^{-1}F_{\max}x_1^{-4}x_2^3x_3 + 1.05x_1x_3 + 2.1x_1,$$
$$K = 8^{-1}Gx_1^4x_2^{-3}x_3^{-1},$$
$$\sigma_p = F_pK^{-1} = 8G^{-1}F_px_1^{-4}x_2^3x_3,$$
$$g_5(\mathbf{x}) = 3x_1 - x_2 \le 0, \tag{40}$$
$$g_1(\mathbf{x}) = 8\pi^{-1}F_{\max}x_1^{-3}x_2[(x_2 - 0.25x_1)(x_2 - x_1)^{-1} + 0.615x_1x_2^{-1}] - S \le 0. \tag{41}$$

Since $g_5(\mathbf{x})$ in (40), it is clear that $x_2 - x_1 > 0$, and $g_1(\mathbf{x})$ in (41) can be transformed by multiplying $(x_2 - x_1)$ on both sides as follows.

$$g_1(\mathbf{x}) = \pi^{-1}F_{\max}(8x_1^{-3}x_2^2 + 2.92x_1^{-2}x_2 - 4.92x_1^{-1}) - S(x_2 - x_1) \le 0, \tag{42}$$
$$g_2(\mathbf{x}) = 8G^{-1}F_{\max}x_1^{-4}x_2^3x_3 + 1.05x_1x_3 + 2.1x_1 - l_{\max} \le 0, \tag{43}$$
$$g_6(\mathbf{x}) = 8G^{-1}F_px_1^{-4}x_2^3x_3 - \sigma_{pm} \le 0, \tag{44}$$
$$g_7(\mathbf{x}) \text{ can be eliminated because of } \sigma_p + \frac{F_{\max} - F_p}{K} + 1.05(x_3 + 2)x_1 - l_f = 0,$$
$$g_8(\mathbf{x}) = \sigma_w - 8G^{-1}(F_{\max} - F_p)x_1^{-4}x_2^3x_3 \le 0. \tag{45}$$

Then, the original formulation can be transformed as follows.

$$\text{Min } 0.25\pi^2x_1^2x_2x_3 + 0.5\pi^2x_1^2x_2$$
s.t. (42)–(45), (38)–(40).

The discrete values of x_1 (wire diameter) and x_2 (outside diameter) are many and diverse. For this reason, the problem will need a lot of space to show the possible values. This example simplifies the possible values that x_1 is a discrete variable with a discreteness of 0.002 and x_2 is a discrete variable with a discreteness of 0.02.

This example uses CPLEX 11 [20] as the MIP solver to solve this larger real-world problem, based on the proposed method and Li and Lu's method. The computational results, shown in Table 8, demonstrate that the proposed method is more efficient computationally than Li and Lu's method for solving this PFDF problem.

Table 8
Computation results for Example 5 using CPLEX 11.

Item	Li and Lu's method	Proposed method
CPU time (s)	1452.81	7.34
No. of simplex iterations	16 444 362	111 964
No. of 0–1 variables	22	22
No. of constraints	2295	288
Solution (x_1, x_2, x_3)	(0.287, 1.3, 8)	
Objective value	2.6421	

5. Conclusions

An efficient logarithmic method has been proposed for handling free-sign discrete signomial (PFDF) terms. The model linearizes PFDF terms directly into a set of linear inequalities without any redundant transformation. Moreover, it only requires logarithmic numbers of binary variables and constraints. Numerical examples show that the novel method for linearizing PFDF terms has a logarithmic number of binary variables and constraints. The computational results demonstrate that it significantly outperforms current methods, especially when the scale of the problem is large.

Acknowledgments

This work was supported by the National Science Council of Taiwan under grants NSC 100-2221-E-030-009-. The authors wish to thank the Area Editor, the Associate Editor, and the anonymous referees for providing insightful comments and suggestions, which have helped us to improve the quality of the paper.

References

[1] A. Paoluzzi, Geometric Programming for Computer Aided Design, John Wiley & Sons, Hoboken, 2003.
[2] S.P. Boyd, S.J. Kim, D.D. Patil, M.A. Horowitz, Digital circuit optimization via geometric programming, Operations Research 53 (2005) 899–932.
[3] H. Cheng, S.C. Fang, J.E. Lavery, A geometric programming framework for univariate cubic L1 smoothing splines, Annals of Operations Research 133 (2005) 229–248.
[4] M. Chiang, Geometric Programming for Communication Systems, Now Publishers Inc., Boston, 2005.
[5] A. Lundell, Transformation techniques for signomial functions in global optimization, Ph.D. Dissertation, Åbo Akademi University, 2009.
[6] Y. Xiong, S.S. Rao, Fuzzy nonlinear programming for mixed-discrete design optimization through hybrid genetic algorithm, Fuzzy Sets and Systems 146 (2004) 167–186.
[7] S.L. Rosen, C.M. Harmonosky, An improved simulated annealing simulation optimization method for discrete parameter stochastic system, Computers and Operations Research 21 (2005) 359–367.
[8] C.A. Floudas, P.M. Pardalos, State of the Art in Global Optimization: Computational Methods and Applications, Kluwer Academic Publishers, Boston, 1996, pp. 257–306.
[9] C.D. Maranas, C.A. Floudas, Global optimization in generalized geometric programming, Computer and Chemical Engineering 21 (1997) 351–370.
[10] C.A. Floudas, P.M. Pardalos, C.S. Adjiman, W.R. Esposito, Z.H. Gumus, S.T. Harding, J.L. Klepeis, C.A. Meyer, C.A. Schweiger, Handbook of Test Problems in Local and Global Optimization, Kluwer Academic Publishers, Boston, 1999, pp. 85–105.
[11] C.A. Floudas, Deterministic Global Optimization: Theory, Methods and Application, Kluwer Academic Publishers, Boston, 2000, pp. 257–306.
[12] R. Pörn, I. Harjunkoski, T. Westerlund, Convexification of different classes of non-convex MINLP problems, Computer and Chemical Engineering 23 (1999) 439–448.
[13] R. Pörn, K.M. Björk, T. Westerlund, Global solution of optimization problems with signomial parts, Discrete Optimization 5 (2008) 108–120.
[14] J.F. Tasi, M.H. Lin, An optimization approach for solving signomial discrete programming problems with free variables, Computer and Chemical Engineering 30 (2006) 1256–1263.
[15] H.L. Li, H.C. Lu, Global optimization for generalized geometric programs with mixed free-sign variables, Operations Research 57 (2009) 701–713.
[16] S.M. Henry, Tight polyhedral representations of discrete sets using projections, simplices, and base-2 expansions, Ph.D. Dissertation, Clemson University, 2011.
[17] H.M. Markowitz, A.S. Manne, On the solution of discrete programming-problems, Econometrica 25 (1957) 84–110.
[18] G.B. Dantzig, On the significance of solving linear-programming problems with some integer variables, Econometrica 28 (1960) 30–44.
[19] J.P. Vielma, G. Nemhauser, Modeling disjunctive constraints with a logarithmic number of binary variables and constraints, Mathematical Programming. Series A 128 (2011) 49–72.
[20] ILOG CPLEX, Version 11, IBM ILOG, Armonk, NY, 2008.
[21] E. Sandgren, Nonlinear integer and discrete programming in mechanical design optimization, Journal of Mechanical Design 112 (1990) 223–229.
[22] LINGO Release 11, Lindo System Inc., Chicago, 2008.
[23] P. Tian, J. Ma, D.M. Zhang, Non-linear integer programming by Darwin and Boltzmann mixed strategy, European Journal of Operational Research 105 (1998) 224–235.